BYOP PROPOSAL

Project Domain : Computer Vision, Augmented Reality

Abhivansh Gupta
23112004
Chemical Engineering
abhivansh_g@ch.iitr.ac.in
+91 8302905563

20 October 2024

Problem Description:

The idea for this project stems from a common challenge in the world of 3D object modelling: how to efficiently create accurate 3D models from 2D images. Traditionally, 3D object reconstruction has been a labor-intensive process requiring specialized tools, expertise in 3D modelling, and a significant amount of manual effort. This problem often arises in industries like augmented reality (AR), virtual reality (VR), gaming, architecture, and e-commerce, where high-quality 3D representations are needed but not easily available.
I found that while the demand for realistic 3D models continues to grow, there is often limited access to datasets or equipment that can capture 3D data directly (such as LiDAR). Relying solely on 2D images for accurate 3D reconstruction presents its own set of challenges, especially when the object of interest is complex, occluded, or viewed from limited angles. This led me to question how AI, particularly **deep learning & generative AI algorithms** could be used to automate and enhance this process, making 3D model creation more accessible and scalable.

Problem Breakdown:

Aim:

To build a generative model that can reconstruct high-quality 3D objects from a single 2D image. The project involves learning the underlying 3D structure of objects and generating plausible 3D models based on the input image. Once the 3D object is reconstructed, rendering it in real-time in an AR environment using techniques from augmented reality.

Challenges:

Implementing architectures like Deep Convolution Generative Adversarial Networks (DCGANs) & Variational Auto-encoders (VAEs) [given I'll be testing using both these methods] in combination with 3D representation techniques such as voxels, point clouds, or mesh-based models.
Handling occlusions and recovering depth information from just a single 2D image.
Training on the ShapeNet/Pix3D dataset, which provides 3D object models with corresponding 2D projections.
Balancing between generating high-resolution, realistic 3D models and maintaining training stability.
AR integration of the 3D constructed objects as it is a majority of a Dev. Part, which I'll be learning to deploy or host.

Tech Stack/Tools/Algorithms that will be used for this project are : Deep Convolution Generative Adversarial Networks, Variational Auto-encoders, ARCore (by Google), Point Clouds, Voxels.

Dataset:

I'll be using Pix3D Dataset that is publicly available. It consists of diverse image-shape pairs with pixel-level 2D-3D alignment.

Link to the dataset : https://github.com/xingyuansun/pix3d

Workflow for 3D Object Reconstruction Using Pix3D:

Data Preparation:

The Pix3D dataset provides both 2D images and their corresponding 3D representations (e.g., mesh files, voxel grids). 2D Images will be used as input to model. We can choose to work with either voxel grids, point clouds, or meshes, depending on the output format your model generates.

Model Architecture:

- **For Voxels**:
  - Will be modifying the DCGAN generator to produce 3D voxel grids as output.
  - The genretor will take 2D images and map them to a 3D voxel grid that represents the object.
  - The discriminator will now take real and generated voxel grids and classify them as real or fake.

- **For Point Clouds**:
  - You might modify the generator to produce 3D point clouds (i.e., sets of 3D points that represent the surface of the object).
  - The discriminator will evaluate whether the generated point clouds correspond to the real object from the dataset.

Training the Model:

- **Input**: 2D images from Pix3D.
- **Output**: 3D voxels (or point clouds, or meshes) from the corresponding 3D data of Pix3D.
- **Loss Function**:
  - Using a typical GAN loss (adversarial loss) where the discriminator learns to classify real and generated 3D outputs.

Evaluation

- Metrics for evaluation include:

    - **Voxel/point cloud overlap**: How well the generated object matches the ground truth.
    - **IoU (Intersection over Union)**: For voxel-based representations.
    - **Chamfer Distance**: For point clouds.

AR Integration:

- Exporting the 3D object generated by the model in a format suitable for AR visualisation (e.g., **OBJ**, **GLTF**, **FBX**).
- Tools like **Google Scene Viewer** to place the 3D object in an augmented reality environment.

TIMELINE FOR THE PROJECT

Week 1: Research and Planning

- **Research State-of-the-Art Approaches**:
    - Exploring various 3D reconstruction techniques, such as voxel-based methods, point clouds, and mesh-based methods.
    - Studying advanced generative models (GANs, VAEs) and their applications in 3D reconstruction.
    - Reading key papers like "**3D-R2N2**", **Pix2Vox**, **AtlasNet**, or newer Transformer-based models for 3D generation.
    - Studying the **ShapeNet** and **Pix3D** datasets to understand the scope of the data and how they are structured.
- **Define the Scope and Approach**:
    - Selecting the type of 3D representation to generate (voxel grids, point clouds, meshes, etc.).
    - Deciding whether to use a single generative model or a combination (e.g., GAN + VAE).
    - Formulating key performance metrics (e.g., Intersection over Union (IoU), Chamfer Distance).

Week 2: Data Preprocessing and Setup & Initial Model Development

- **Downloading and Preparation of Dataset**:
    - Downloading the **ShapeNet** and/or **Pix3D** datasets.
    - Preprocessing the data by converting 3D object files into the desired representation (voxel grids, point clouds, meshes).
    - Augmenting the 2D images if necessary (scaling, rotations) for better generalization.
    - Setting up data pipelines to load and preprocess the 2D and 3D data efficiently.

- **Baseline Model Setup**:
    - Startihg implementing a basic model architecture (e.g., a simple VAE for 3D voxel reconstruction from 2D images).
    - Training this basic model to verify pipeline is functional.
- **Model Architecture Development**:
    - Implementing more sophisticated generative models like 3D Convolutional Auto-encoders, GANs, or VAEs.
    - If using GANs, developing both the generator (to predict 3D shapes) and discriminator (to classify between real and generated shapes).
    - Adding spatial/positional encoding or use Transformer layers if exploring more advanced architectures.
- **Training the Model**:
    - Begin training the model on a subset of the dataset to ensure everything runs smoothly.
    - Monitoring basic performance metrics and visualising intermediate outputs to validate that the 3D object reconstructions are plausible.
- **Loss Functions and Metrics**:
    - Implementing key loss functions like **Reconstruction Loss**, **Adversarial Loss** (for GANs), or **Chamfer Distance** for point clouds.
    - Setting up evaluation metrics like IoU for voxel reconstructions or surface distance metrics for meshes.

Week 3: Model Tuning and Experimentation

- **Hyperparameter Tuning**:
    - Performing hyper-parameter tuning (batch size, learning rate, optimiser, latent dimension size) to optimize performance.
    - Experimenting with different network architectures (e.g., deeper encoders/decoders, larger latent spaces).
- **Data Augmentation and Regularization**:
    - Adding 2D and 3D data augmentation techniques (e.g., noise addition, transformations) to improve generalization.
    - Exploring regularization techniques like dropout or weight normalisation to avoid overfitting.
- **Intermediate Testing**:
    - Testing my model on unseen images from the dataset to evaluate generalization to new inputs.
    - Visualizing reconstructions for qualitative assessment: compare real vs generated 3D shapes.

Week 4: Fine-tuning and Advanced Features

- **Refining the Model**:

  - Improve specific issues observed during testing (e.g., poor detail in reconstructions or artefacts in 3D shapes).
  - Experimenting with multi-view learning (using multiple 2D images of the same object from different angles to improve 3D reconstructions).
- **Deploying on Edge Devices (If time permits me)**:
  - If real-time reconstruction is possible within time, optimizing the model to run efficiently on GPUs or edge devices.
- **Multiview Consistency**:
  - Working on ensuring multi-view consistency if generating from multiple 2D images, making sure the 3D shapes match across different views.
